Autonomous Human Activity Classification from Wearable Multi-Modal Sensors

Yantao Lu, *Member, IEEE,* Senem Velipasalar, *Senior Member, IEEE*

***Abstract*—There has been significant amount of research work on human activity classification relying either on Inertial Measurement Unit (IMU) data or data from static cameras providing a third-person view. There has been relatively less work using wearable cameras, providing first-person or egocentric view, and even fewer approaches combining egocentric video with IMU data. Using only IMU data limits the variety and complexity of the activities that can be detected. For instance, the sitting activity can be detected by IMU data, but it cannot be determined whether the subject has sat on a chair or a sofa, or where the subject is. To perform fine-grained activity classification, and to distinguish between activities that cannot be differentiated by only IMU data, we present an autonomous and robust method using data from both wearable cameras and IMUs. In contrast to convolutional neural network-based approaches, we propose to employ capsule networks to obtain features from egocentric video data. Moreover, Convolutional Long Short Term Memory framework is employed both on egocentric videos and IMU data to capture the temporal aspect of actions. We also propose a genetic algorithm-based approach to autonomously and systematically set various network parameters, rather than using manual settings. Experiments have been conducted to perform 9- and 26-label activity classification, and the proposed method, using autonomously set network parameters, has provided very promising results, achieving overall accuracies of 86.6% and 77.2%, respectively. The proposed approach, combining both modalities, also provides increased accuracy compared to using only egovision data and only IMU data.**

***Index Terms*—Activity classification, genetic algorithm, capsule network, egocentric, egovision, IMU data**

I. Introduction

Many approaches have been proposed to perform human activity classification from different sensors. Most of the existing methods rely either on Inertial Measurement Unit (IMU) data [1][2][3][4][5][6][7][8] or data from static cameras in the environment providing a third-person view [9][10][11][12][13].

Mannini and Sabatini [1] [2] use IMU data to classify activities of sitting, standing, lying down, walking, running, climbing stairs and cycling. Bayat et al. [7] also employ IMU data to classify activities, such as dancing, going up and down the stairs, slow and fast walking and running, and compare different classifiers. Ordóñez and Roggen [8] use accelerometer and gyroscope data, and employ convolutional and Long Short Term Memory (LSTM) recurrent units for activity recognition. Even though the systems using only IMU data are computationally efficient, they are limited in terms of the variety and complexity of the activities that they can detect. They also cannot provide enough context. For example, IMU data can help to detect a sitting activity, but cannot help determine the type of furniture the subject sits on or the environment the subject is in. Having data from a camera sensor fills in these blanks by providing abundant information about surroundings and the objects with which the subject is interacting.

On the other hand, most of the camera-based activity detection works use static cameras watching the subjects, and thus providing a third-person view. Karpathy et al. [9] merge different convolutional neural networks (CNN) for large-scale video classification, and present results on the UCF-101 Action Recognition Dataset [14]. Donahue et al. [10] use long-term recurrent network together with CNNs, and also evaluate on UCF-101 dataset. Instead of using 2D CNN and LSTM, different approaches have been presented using 3D CNNs for learning spatiotemporal features [11]. Montes et al. [12] use a 3D CNN together with LSTM to achieve temporal activity detection in untrimmed videos. Instead of using LSTM, Buch et al. [13] use 3D CNN together with GRUs (Gated Recurrent Unit) on videos from a third-person view. Heilbron et al. [15] present ActivityNet, which is a large-scale video benchmark for human activity understanding, and propose a method based on 3D CNNs. In this video set, the majority of videos are not egocentric.

Wearable sensors are becoming more and more ubiquitous in our lives. Potential applications of activity recognition from egocentric videos include life logging, video diaries and video summarization, health care, elderly care, personal assistance to users or caregivers, navigation and assistance for the visually impaired, robotics, human-human and human-robot interaction and law enforcement. Several surveys [16][17] have been published describing various techniques for activity classification, and heart rate and sleep sensing by wearable sensors. Compared to static cameras, there have been relatively less work as well as much fewer datasets focusing on wearable cameras, egovision and combination of wearable camera data with other sensor modalities.

Existing works focusing on egocentric videos differ in terms of the types of objects and activities that they detect. There have been methods focusing only on hand detection from egocentric videos [18]. Other approaches employing egocentric video data either classify the activities observed

Yantao Lu and Senem Velipasalar are with the Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY, 13244 USA e-mail: {ylu25, svelipas}@syr.edu.

The information, data, or work presented herein was funded in part by National Science Foundation (NSF) under Grant 1739748, Grant 1816732 and by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0000940. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

by the camera [19][20] or the activities of the person wearing the camera [21][22][23][24][25][26]. Ryoo and Matthies [20] present a method to recognize what activities others are performing to the observer or the person or robot wearing the camera. Pirsiavash and Ramanan [23] presented a dataset of egocentric videos covering Activities of Daily Living (ADL), and reported 40.6% accuracy over 18 classes. McCandless and Grauman [24] presented a method for activity recognition by learning the spatio-temporal partitions. They used the same ADL dataset, and reported 38.7% accuracy over 18 classes. Lu and Grauman [27] presented a method for story-driven video summarization and tested it on the ADL dataset [23]. Moghimi et al. [26] presented a method for activity detection using RGB-D egocentric videos. Nguyen et al. [28] provided a survey and review of the egocentric vision systems for the recognition of activities of daily living. It was concluded that the performance of current systems is far from satisfactory.

The aforementioned approaches, which focus on egocentric videos, are based on only a single sensor modality, namely the camera. There have been even fewer approaches that combine egocentric video data with IMU data [29][30][31][32][33]. Zhan et al. [30] use a smartphone attached on top of safety goggles to collect video and 3-axis acceleration data. They use optical flow vectors from camera data and classify 12 activities, including walking, going upstairs/downstairs, sitting, standing, drinking and writing. Windau and Itti [31] also use both IMU and camera data from a prototype eyeglass setup. They extract GIST features from camera data to perform indoor/outdoor classification. They report 81.5% accuracy for classifying 20 activities including lying down, walking, jogging, biking, running, playing cello, playing piano, computer work, folding laundry and driving car. However, both of these methods still focus on activities that can be classified by only accelerometer data. In other words, they do not perform detection of objects in the scene, and do not focus on activity types involving interactions with different types of objects, which cannot be classified by only accelerometer data. In earlier work, Spriggs et al. [32] used the CMU Multi-Modal Activity (CMU-MMAC) database [34], and presented a method for temporal segmentation and activity classification, focusing on recipe preparation, by extracting the GIST features from the egocentric video data. They reported 57.8% as the highest performance.

Different from the eyeglass setup, Conti et al. [29] employed the various sensors on a smartwatch to perform context classification over only 5 classes (morning preparation, walking outdoors, public transportation, in the car and in the office).

It should be emphasized that many activities can be very close to each other in the "activity space", in other words, can be very similar, such as using a spoon versus using a fork. In this case, adding another sensor modality, namely the camera, and detecting objects become even more important to identify activities involving interactions with various-sized objects. The problem gets much harder for fine-grained classification of activities. As mentioned above, most of the existing work does not focus on fine-grained activity classification. On the other hand, the relatively small number of existing works on fine-grained classification have reported lower accuracies.

In this paper, we present an autonomous and robust method to perform fine-grained activity classification and distinguish between activities, which cannot be differentiated by only IMU data, by using data from multi-modal wearable sensors including a camera. In order to analyze and extract features from egocentric video data, we propose a new model architecture, and employ a capsule network [35], in contrast to many CNN-based approaches. In addition, Convolutional LSTM framework is employed both on egocentric videos and IMU data to capture the temporal aspect of actions, which span a time window. In other words, we use multiple capsule networks for consecutive images.

The choice of parameters, and the design of the network architecture are important factors affecting the performance of deep learning methods. Many researchers proposed different CNN architectures [36], [37], [38], [39], [40], [41], [42] to achieve higher accuracy. However, building the neural network structure still heavily depends on manual trial and error, and empirical results. Considering that there are many design and parameter choices, it is not possible to cover every possibility, and it is very hard to find the optimal structure. Moreover, the hyper-parameters in the training phase also play an important role on how well the model will perform. Likewise, these parameters are also tuned manually in an empirical way most of the time. Thus, rather than using trial and error for various parameter combinations, we also propose a genetic algorithm (GA)-based approach to autonomously and systematically set the various parameters of our network architecture.

In our preliminary work [43], we explored using egocentric video, IMU data and recurrent Capsule Networks for activity classification. The work proposed in this paper is different and improved compared to our previous work [43] in multiple ways including the following: (i) for the work in [43], we used manually set values for all the network parameters. In contrast, in this paper, we propose a GA-based approach to autonomously and systematically set various network parameters, rather than using manual and empirical settings; (ii) the experiments in [43] were performed for only 6-label classification. In this paper, we provide a much more comprehensive evaluation by performing classification for 9 as well as 26 activities; (iii) in this paper, we provide a more detailed description of the proposed method and a comparison between the performances obtained with the manually preset network parameters, and the parameters determined by our proposed GA-based approach.

Experiments have been performed on the CMU-MMAC dataset to perform 9- and 26-label classification, and the proposed method, using autonomously set network parameters, has provided very promising results, achieving overall accuracies of 86.6% and 77.2%, respectively. We also used each sensor modality alone, and obtained their individual accuracies, showing that the proposed approach, combining both modalities, provides increased accuracy compared to using only egovision data and only IMU data.

The rest of this paper is organized as follows: The proposed approach is described in detail in Section II. Experimental results are presented in Section III, and the paper is concluded in Section IV.

II. Proposed Method

We present a new model architecture to process first-person, also known as egocentric, images and IMU data. The proposed architecture can be seen in Fig. 1. It is composed of our proposed recurrent CapsNet (for processing images), an LSTM network (for processing IMU data), and fully connected layers. In addition, we also propose and apply a GA-based approach to autonomously and simultaneously optimize multiple parameters of our network architecture. These parameters are shown in parentheses with red color in Fig. 1. For instance, the parameters for the fully connected layers and the primary capsules are examples of the parameters autonomously set by our proposed approach. More details about these parameters will be provided in Sec. II-A.

Sabour et al. [35] introduced the Capsule Networks (CapsNets) to explore spatial relationships between features, and reported state-of-the-art performance on the MNIST database. CapsNets [35] were used for image classification on individual images, whereas our goal is to perform fine-grained activity classification from video data. Thus, in this paper, instead of using a single image with the original CapsNet, we propose a Recurrent CapsNet (RecCapsNet), which takes a sequence of images as input. We implement a 2D Convolutional LSTM (convLSTM) [44] layer to extract features and capture the temporal aspect. For robustness, we use multiple digit/class layers instead of using only a single digit layer as was done in [35]. In order to prevent gradient vanishing, we remove the squash function for digit layers and implement ReLu activation function instead.

As seen in Fig. 1, 16 consecutive images are passed through a 2D convolutional layer separately. The size of each input image is 36×36. Then, the output for each image is sent to multiple primary capsules, the number of which is determined by our GA. When 16 consecutive images are formed, 50% overlap is used throughout the video. The number of convolutional units for each primary capsule is also determined by the GA. The output from the primary capsule layer is then sent through two digit/class layers, whose parameters are set by the GA. We then apply a Convolutional LSTM layer, followed by a fully connected (FC) layer, for the analysis of the egocentric video data.

For the decoder part, we apply 16 sub-decoders to each image frame. Each sub-decoder has the same structure with the decoder of the original CapsNet except the sigmoid output is 1296 (36×36). In other words, the decoders are implemented to generate the same size as the input images. Given the ground truth label, the decoder regenerates a 16 frame image sequence which has the same size as the image input.

As for the IMU data, similar to the images, data from 16 consecutive time frames are used. Each of the 16 IMU data vectors has 36 components obtained by concatenating data from the four IMU sensors. Each IMU sensor contributes nine entries from accelerometer, gyroscope and magnetometer measurements. The time stamps are provided for camera and IMU data in the CMU-MMAC dataset. To align the camera and IMU data, for a given camera image, the IMU time stamp that is closest to the camera time stamp is found. The IMU data is fed into an LSTM model to extract feature vectors, which are then sent to the FC layer(s). The outputs of the fully connected layer for video data and the fully connected layer for the IMU data are concatenated, and the concatenated features are then fed into another FC layer for classification. The number of neurons for this FC layer is also set by the GA, and it is denoted by $Para\_FC\_merge$ and shown in red in Fig. 1. This FC layer is followed by a softmax classifier. The output of the model is the confidence scores for each class proposal.

Next, in Sec. II-A, we will describe the details of the algorithm that we propose to refine the various parameters of this architecture by using a Genetic Algorithm.

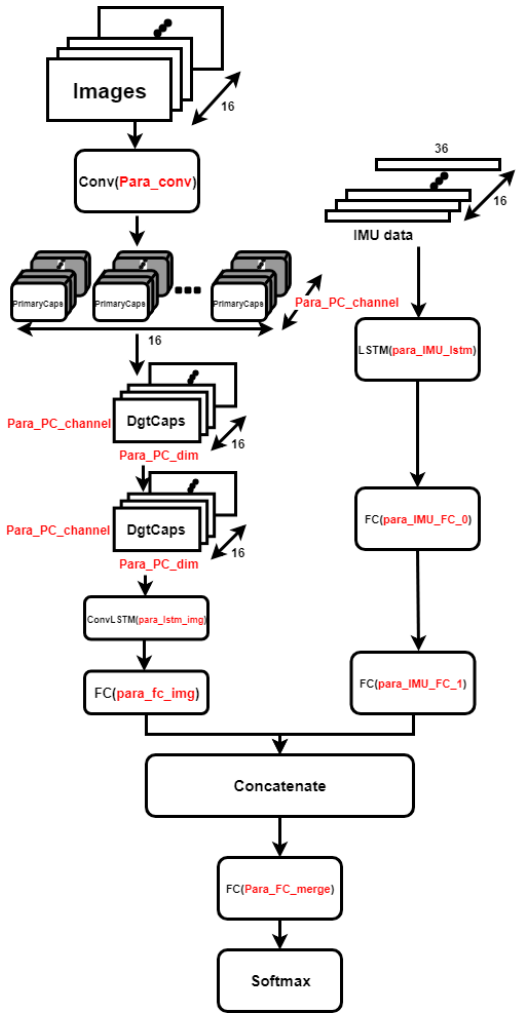


Fig. 1: Details of the proposed architecture.

A. Autonomously and Simultaneously Refining the Network Parameters

The overall structure of the proposed method to refine the network parameters is shown in Fig. 2. In this approach, a GA is used to make a decision from a set of discrete choices. The decisions by the GA include the choice of the activation function and the optimizer; whether or not to use batch normalization, dropout and max-pooling; the number

TABLE I: Parameters Autonomously Chosen by the GA

optimizers	{“adam”, “rmsprop”, “adagrad”, “adadelta”}
activation functions	{“relu”, “leaky relu”, “sigmoid”, “tanh”}
batch normalization	{True, False}
dropout	{True, False}
max pooling	{True, False}
kernel size	{3, 6, 9}
kernel stride	{1, 2, 3}
number of conv filters	{32, 64, 128 ... 512}
number of dense neurons	{32, 64, 128, 256}
number of lstm units	{16, 32, 64 ... 256}
dimension of capsules	{2, 4, 8, 16}
number of primary channels	{16, 32, 64}
number of conv layers	{3,6}
number of dense layers	{1,3}
number of LSTM layers	{1,3}

of convolutional layers and dense layers; the kernel size and stride, the number of LSTM units, etc. The complete set of parameters together with the discrete set of values that they can take are shown in Table I.

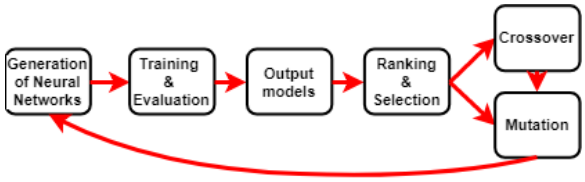


Fig. 2: The structure of the proposed Genetic Algorithm

The parameter set of the network i for the GA has the following form:

$$\begin{aligned} p_i^{GA} =&[prm_i^{conv}, prm_i^{PC}, prm_i^{LSTM_{img}}, prm_i^{FC_{img}}, \\ & prm_i^{LSTM_{IMU}}, prm_i^{FC^0_{IMU}}, prm_i^{FC^1_{IMU}}, \\ & prm_i^{FC_{merge}}, prm_i^{O}] \\ & i \in \{1, 2, ..., n_m\} \end{aligned} \tag{1}$$

where PC and FC denote primary capsule and fully connected layer, respectively, n_m is the number of network models in the population and

$$\begin{aligned} prm_i^{conv} =&[1/0 \text{ (conv. lyr exists or not)}, \text{no. of filters}, \\ & \text{kernel size}, \text{stride}, \text{activation func.}, \\ & 1/0 \text{ (for batch norm.)}], \\ & i \in \{1, 2, ..., n_m\}, \end{aligned} \tag{2}$$

$$\begin{aligned} prm_i^{PC} =&[\text{capsule dim.}, \text{num of chan.}, \\ & \text{kernel size}, \text{stride}] \\ & i \in \{1, 2, ..., n_m\}, \end{aligned} \tag{3}$$

$$\begin{aligned} prm_i^{LSTM_{img}} =&[1/0 \text{ (LSTM lyr exists or not)}, \text{no. of units}, \\ & \text{activation func.}] \\ & i \in \{1, 2, ..., n_m\}, \end{aligned} \tag{4}$$

$$\begin{aligned} prm_i^{FC_{img}} =&[1/0 \text{ (dense lyr exists or not)}, \text{no. of neurons}, \\ & \text{activation func.}, 1/0\text{(for dropout)}] \\ & i \in \{1, 2, ..., n_m\}, \end{aligned} \tag{5}$$

$$\begin{aligned} prm_i^{LSTM_{IMU}} =&[1/0 \text{ (LSTM lyr exists or not)}, \text{no. of units}, \\ & \text{activation func.}] \\ & i \in \{1, 2, ..., n_m\}, \end{aligned} \tag{6}$$

$$\begin{aligned} prm_i^{FC_{IMU^{0/1}}} =&[1/0 \text{ (dense lyr exists or not)}, \text{no. of neurons}, \\ & \text{activation func.}, 1/0\text{(for dropout)}] \\ & i \in \{1, 2, ..., n_m\}, \end{aligned} \tag{7}$$

$$\begin{aligned} prm_i^{FC_{merge}} =&[1/0\text{(dense lyr exists or not)}, \text{no. of neurons}, \\ & \text{activation func.}, 1/0\text{(for dropout)}] \\ & i \in \{1, 2, ..., n_m\}, \end{aligned} \tag{8}$$

$$\begin{aligned} prm_i^{O} =&[\text{type of optimizer}] \\ & i \in \{1, 2, ..., n_m\}. \end{aligned} \tag{9}$$

1) Initial Population: The first generation of the networks, N^1, is generated randomly such that $N^1 = \{N_1, N_2, ..., N_{n_m}\}$, where n_m is the number of models. This is done by choosing the values of parameters, in (2) through (9), randomly, from the possible choices. The value of n_m was set to be 10 in our experiments.

2) Evaluation: Each generated network model N_i ($i \in \{1, ..., n_m\}$) is evaluated by the fitness function $f(N_i)$, which is a measure of the accuracy of each model. Models with better performance will return higher values. Thus, $E = \{E_1, E_2, ..., E_{n_m}\}$, will hold the fitness scores $E_i = f(N_i)$, where $i \in \{1, 2, ..., n_m\}$.

3) Selection: In the selection part, t-many top-ranked models are selected from the $sorted(E)$, and r-many models are selected randomly from the rest of the network models. Then, d-many models are dropped in order to prevent over-fitting and getting stuck at a local optimum. The remaining selected models are the parent models (P), which will be used to create new models for the next generation.

4) Crossover and Mutation: Crossover is applied to generate n_m-many child network models from the parents. As opposed to always choosing two parents randomly from the parent pool, we associate a counter C_P with each parent P, and initialize it to zero. This counter is incremented by one each time a parent is used for crossover. First, two parents are selected randomly from the $t + r - d$ many parents. A new ‘child’ network is generated from the parents via crossover, and the counters of the parents are incremented by one. Then, two parents, whose counter is still zero, are selected randomly from the parent pool. Another network is generated from them via crossover, and the counters of the parents are incremented. If there is only one network model left with counter equal to zero, and the number of children is still less than n_m, then this model is chosen as one of the parents, and the other parent is chosen randomly from the rest of the models who have a counter value of one. If there are no more parents left with counter equal to zero, and the number of children

is still less than n_m, then two parents, whose counter is one, are picked randomly, and their counter is incremented to two after crossover. This process is repeated until the number of children models reaches n_m.

The crossover between parent models a and b is performed, as illustrated in Fig. 3, by using a single-point crossover. As seen from equations (1) through (9), there are a total of 33 parameters in each parent vector. An index number ind is picked randomly between 1 and 33. Parameters to the left of ind from the parent a vector and to the right of ind from the parent b vector are selected to compose the child vector. In other words, parameters 1 through ind, and $ind+1$ through 33 are selected from parents a and b, respectively, to form the child vector.

After all the n_m-many child networks are obtained via crossover, the mutation is performed. From each vector, k-many indices are chosen randomly to perform mutation. The values of the parameters corresponding to the chosen indices are randomly changed to one of the possible choices shown in Table I. For instance, if the random number corresponds to the dimension of the capsules, then its value is chosen randomly from $\{2, 4, 8, 16\}$. The value of k was chosen to be 3 in our experiments.

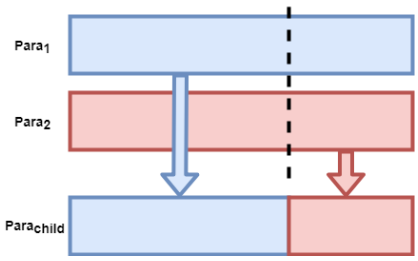


Fig. 3: Crossover process for the GA

Then, the entire process is repeated by using this new population. The pseudo code for GA-based parameter setting is provided in Algorithm 1. After the parameters are set autonomously by the GA-based approach, the network model is generated as described in the pseudo code in Algorithm 2.

Algorithm 1: The Genetic Algorithm

Randomly initialize n_m models for population N^1.
while i^{th} iteration **do**
  Train and evaluate $N_1^i, N_2^i, ..., N_{n_m}^i$ by fitness function $f(N_j^i)$ and obtain scores E.
  Select t top scored networks $N_{top} = N^i(argmax(E))$
  Randomly choose r networks N_{rand} from the rest of population N^i
  Merge N_{top} and N_{rand} and then drop d networks (N_{drop})
  Form $N_{parent} = (N_{top} \bigcup N_{rand}) - N_{drop}$
  Choose parents from N_{parent} for crossover and generate n_m new networks and add them to N^{i+1}
  Perform mutation on k-many elements of vectors in N^{i+1}.
end

Algorithm 2: Network model generation from the GA-set parameters

Input: Genetic representative vector $\mathcal{L}$; vector prototype $p^{GA} = [prm_{video}, prm_{IMU}, prm_{merge}]$ shown in eq.(1);
$prm_{video} = [prm_i^{conv}, prm_i^{PC}, prm_i^{LSTM_{img}}, prm_i^{FC_{img}}]$
$prm_{IMU} = [prm_i^{LSTM_{IMU}}, prm_i^{FC^0_{IMU}}, prm_i^{FC^1_{IMU}}]$
$prm_{merge} = [prm_i^{FC_{merge}}]$
Input: video input shape S_v; IMU input shape S_{IMU};
Output: output model M
$input_{video} \leftarrow Placeholder(shape = S_v)$
for $i = 0$ to $length(input_{video}) - 1$ **do**
  $offset \leftarrow 0$
  $M_i \leftarrow Sequential(input_{video}[i])$
  for $idx = 0$ to $length(prm_{video})$ **do**
    Build layer L_{idx} from $L[offset : offset + len(prm_{video}[idx])]$
    $M_i \xleftarrow{+} L_{idx}$
    $offset+ = length(prm_{video}[idx])$
  end
  $output^i_{video} = M_i(input^i_{video})$
end
$input_{IMU} \leftarrow Placeholder(shape = S_{IMU})$
$M_{IMU} \leftarrow Sequential(input_{IMU})$
for $idx = 0$ to $len(prm_{IMU})$ **do**
  Build layer L_{offset} from $L[offset : offset + length(prm_{IMU}[idx])]$
  $M_{IMU} \xleftarrow{+} L_{idx}$
  $offset+ = length(prm_{IMU}[idx])$
end
$output_{IMU} = M_{IMU}(input_{IMU})$
obtain concatenate layer L_{concat} from $output_{video}$ and $output_{IMU}$
$M_{merge} \leftarrow Sequential(L_{concat})$
for $idx = 0$ to $length(prm_{merge})$ **do**
  Build layer L_{idx} from $L[offset : offset + len(prm_{merge}[idx])]$
  $M_{merge} \xleftarrow{+} L_{idx}$
  $offset+ = len(prm_{merge}[idx])$
end
$output_{merge} = M_{merge}(L_{concat})$
return $Model([input_{video}, input_{IMU}], output_{merge})$

III. Experimental Results

A. Experimental Setup

We have used the CMU-MMAC dataset [34] for the experiments. This dataset contains data from multi-modal sensors monitoring human subjects preparing food. A kitchen was built and 25 subjects were recorded cooking five different recipes, namely brownies, pizza, sandwich, salad, and scrambled eggs. The sensor modalities used for data collection include three high-resolution static cameras, two low-resolution static cameras, one wearable camera, five microphones, and IMUs. In our experiments, we used the egocentric (egovision) camera data (from the wearable camera) and the wired IMU data.

We resized the image frames from the camera to 36×36 pixels, and processed 16 consecutive frames each time with 50% overlapping. We down-sampled the IMU data to make the measuring frequency the same with the egocentric camera (30 Hz). Then, we synchronized/aligned the IMU data with camera data.

We performed two sets of experiments with different number of classes. More specifically, we performed 9-class labeling and 26-class labeling by using 9 and 26 different activity classes, respectively. The activities used for the 9-label classification are:
A_9= {'fridge(open or close)', 'taking/beating eggs', 'pouring into big bowl', 'pouring into cup', 'stirring in a bowl', 'taking bowl', 'taking baking pan', 'taking measuring cup', 'twisting cap (on or off)'}.
Example images for these nine classes can be seen in Fig. 9.

The activities used for the 26-label classification are:
A_{26} = {'closing fridge', 'cracking egg', 'opening brownie bag', 'opening fridge', 'pouring big bowl into a pan', 'pouring brownie bag into a bowl', 'pouring oil into a cup', 'pouring water into a bowl', 'pouring water into a cup', 'putting pan into oven', 'putting cooking spray/pam into cupboard', 'spraying cooking oil', 'stirring in a bowl', 'switching on', 'taking baking pan', 'taking bowl', 'taking brownie box', 'taking eggs', 'taking fork', 'taking big cup', 'taking small cup', 'taking cooking spray', 'twisting cap off', 'twisting cap on', 'walking to the counter', 'walking to the fridge'}.

As can be seen, especially for the 26-class case, the activities involved are very close in the 'activity space', and this fine-grain classification is a very challenging problem.

A total of 10 videos from subjects 07, 08, 09, 12 and 13 (2 videos per subject) have been used for training and testing. Videos from each subject were randomly divided so that 70%, 20%, 10% of the samples were allocated for training, validation and testing, respectively.

We also compared our results with two other works [32][45], which also use the same CMU-MMAC dataset. All the results are presented below in Section III-B.

B. Results and Discussion

As mentioned above, we performed 9-class and 26-class labeling in our experiments. In each scenario, we first performed classification with manually preset network parameters, and then with the parameters determined autonomously by our GA-based approach described above. Preset parameters were obtained by choosing the parameter configuration that resulted in the best performance after multiple trials. For both 9-class and 26-class labeling, the preset parameters (corresponding to equations (1) through (9)) are:

$$[1, 256, 9, 1, 0, 0, 8, 32, 9, 2, 1, 256, 0, 1, 128, 0, 0, 1, 128, 0, 1, 128, 0, 0, 1, 32, 0, 0, 0, 128, 0, 0, 0] \quad (10)$$

The parameters determined autonomously by our proposed GA-based approach are:

$$[1, 32, 6, 3, 0, 1, 16, 32, 3, 1, 0, 32, 0, 0, 128, 1, 0, 0, 64, 1, 1, 64, 0, 0, 0, 256, 1, 0, 0, 32, 0, 0, 0] \quad (11)$$

and

$$[1, 64, 9, 3, 0, 1, 8, 64, 6, 2, 0, 256, 1, 0, 64, 1, 0, 0, 64, 1, 1, 256, 0, 0, 0, 128, 1, 0, 0, 64, 0, 0, 0] \quad (12)$$

for 9-class and 26-class classification, respectively. For instance, in both cases, the GA-based approach results in less number of filters for the convolutional layers (32 and 64 instead of 256), and less number of neurons for the fully connected merge layer. The overall accuracies from these experiments are summarized in Table II, wherein the accuracy is the ratio of all correctly classified instances to the total number of instances. As can be seen, when we use our proposed GA-based approach to autonomously set the various parameters of the network, this provides higher accuracy for both 9-class and 26-class labeling. Thus, the remainder of the results are presented for when the parameters are set with our GA-based approach.

TABLE II: Overall accuracies for the 9- and 26-class labeling with and without using the proposed GA-based parameter setting

	9-class		26-class	
	Preset parameters	GA-based parameters	Preset parameters	GA-based parameters
Accuracy	84.2%	**86.6%**	75.7%	**77.2%**

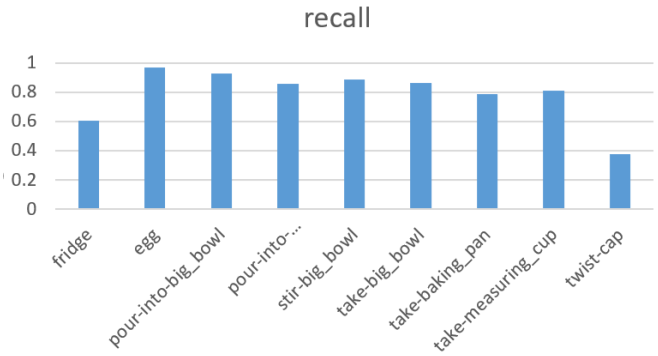


Fig. 4: The recall values for each of the 9 classes.

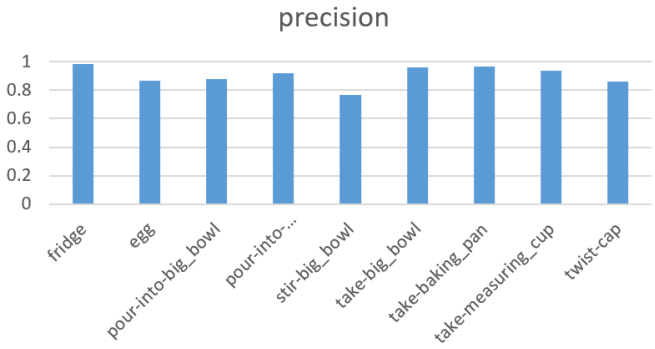


Fig. 5: The precision values for each of the 9 classes.

The recall and precision values for each class, for the 9-class case, are shown in Figures 4 and 5, respectively. For the 26-class case, the recall and precision values for each class are shown in Figures 6 and 7, respectively. The confusion matrices

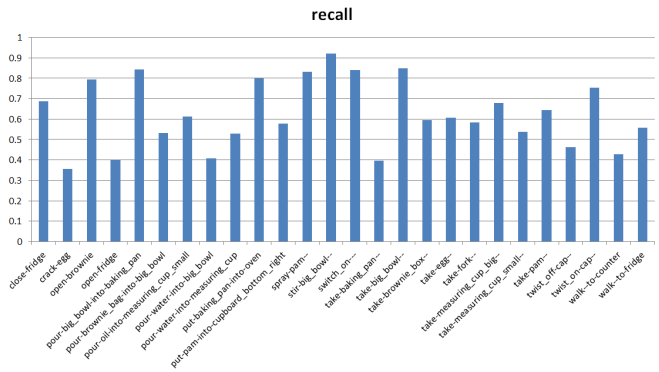


Fig. 6: The recall values for each of the 26 classes.

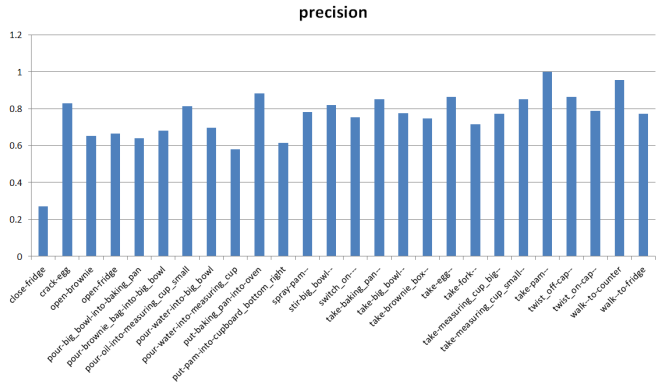


Fig. 7: The precision values for each of the 26 classes.

for the 9-class and 26-class activity classification are shown in Figures 8(a) and 8(b), respectively. As can be seen from the precision-recall graphs and the confusion matrices, when subjects interact with larger objects, and subject movements are faster, higher accuracy is achieved compared to the slower movements and interacting with smaller objects. For instance, it is harder to detect 'twisting cap on' and 'twisting cap off' actions, since the cap is always occluded by hand in the camera view. As another example, actions such as 'cracking egg' are also harder to classify, since the egg is much smaller than the bowl.

In addition, as expected, higher overall precision and recall rates are achieved for 9-class labeling, since activities are much closer to each other and harder to differentiate for the 26-class labeling case. In Fig. 10, we show example images for the activities that are confused with each other in the 26-class labeling case (based on the confusion matrix in Fig. 8(b)). These images illustrate once more the difficulty of performing very fine-grained activity classification. The first row shows taking a small cup (on the left) vs. big cup (on the right). The second row shows walking to the fridge (on the left) vs. closing the fridge (on the right), and finally the third row shows pouring into pan (on the left) vs. putting the pan into the oven (on the right). As can be seen, these are very similar looking activities, and the proposed approach still provides very promising results for the 26-class labeling. More discussion and comparison will be provided below.

After setting the various network parameters by our GA-based approach, we then performed a comparison of our proposed Rec-CapsNets method with using VGG16 features. For this comparison, instead of employing the proposed Rec-CapsNet, we extracted image features from 16 consecutive image frames by using the convolutional layers of the CNN-based VGG16 [46] without the top layers. We also used CapsNet on individual frames. We used the same dataset splitting, described above, for all compared methods. The results are summarized in Table III for 9-label classification. As can be seen, using our proposed RecCapsNet provides a higher accuracy than using the VGG16 features. Moreover, to show the improvement provided by using multiple sensor modalities, we also obtained results by using each sensor modality by itself, namely by using only IMU data and only camera data. As can be seen in Table III, the proposed approach provides 29.07%, 20.29% and 19.16% increase in accuracy compared to using only IMU data, only egocentric camera data with VGG16 features and only egocentric camera data with CapsNet features, respectively.

The above comparison was performed for 26-label classification as well, and the results are summarized in Table IV. Similar to the 9-class case, using our proposed RecCapsNet together with LSTM on IMU data provides a higher accuracy than using the VGG16 features. In addition, the proposed approach provides 28%, 19.5% and 25.2% increase in accuracy compared to using only IMU data, only egocentric camera data with VGG16 features and only egocentric camera data with CapsNet features, respectively. For Tables III and IV, the parameters used for each approach are as follows:

$$\begin{aligned} \text{LSTM (for IMU data): } & \text{LSTM (128)} \rightarrow \text{LSTM(64)} \rightarrow \\ & \text{FC(128)} \rightarrow \text{FC(64)} \\ \text{VGG16: } & \text{parameters from [46]} \\ \text{CapsNet: } & \text{parameters from [35]} \end{aligned}$$

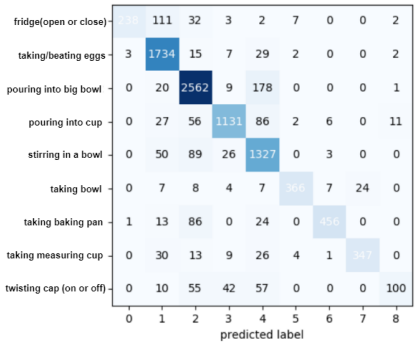
For the proposed method (in Tables III and IV), we used the parameters in equations (11) and (12), respectively, which were autonomously set by our GA-based approach.

Since this is fine-grained classification, using multiple modalities provides a significant increase in performance compared to single-modality results.

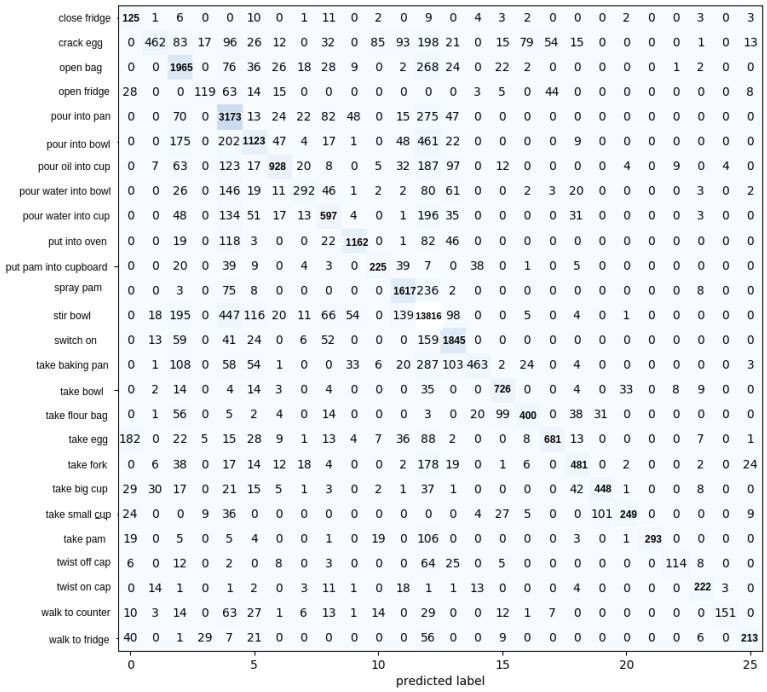
TABLE III: Accuracy rates from different modalities and approaches for 9-label classification

Sensor Modality	Method	Accuracy
IMU only	LSTM	57.57%
Camera only	VGG16	66.35%
	CapsNet	67.48%
Camera and IMU	VGG16 & LSTM	82.97%
	RecCapsNet & LSTM (Proposed)	**86.64%**

We also compared our results with two other works [32][45], which also use the same CMU-MMAC dataset. Soran et al. [45] does not use IMU data, and either employ egocentric camera data or combine egocentric camera data with static camera data. From only the egocentric camera data, Soran et al. [45] report an accuracy of 37.92% for 28 classes. With our approach, when we exclude the IMU data, we obtain an accuracy of 67.48% and 52% for 9-class and 26-class



(a) 9-class labeling



(b) 26-class labeling

Fig. 8: Confusion matrices showing the correct versus predicted classes together with the number of instances of each activity for (a) 9-class and (b) 26-class activity classification.

labeling, respectively. Thus, the proposed approach provides much higher performance.

Spriggs et al. [32], on the other hand, report an accuracy of 57.8% over 29 classes when using IMU data and egocentric camera data together. Our accuracy over 26 classes listed in 8(b) is 77.2%. In order to make the comparison more commensurate, we performed an additional experiment. More specifically, we have trained and tested our proposed method with the same 29 classes used in [32]. The accuracy we obtained is 83.03% for the 29-class labeling. These results are summarized in Table V.

Overall, our proposed method provides a significant improvement without relying on the static cameras watching the targets, which could also be important to alleviate the privacy concerns. In addition, using the proposed GA-based approach not only provides a way to systematically set the network parameters, but also improves the performance further compared to using the manually set parameters.

IV. Conclusion

We have presented a robust and autonomous method to perform fine-grain activity classification by leveraging data from multiple sensor modalities, more specifically egocentric video and IMU sensor data from wearable devices. In contrast to many CNN-based approaches, we have proposed to use a capsule network to obtain features from egocentric video data. Instead of using a single CapsNet, multiple CapsNets are employed for consecutive images, and then a convolutional LSTM is used to build a recurrent CapsNet. The LSTM framework is employed both on IMU data and egocentric camera data to capture the temporal aspect of actions, which span a time window. Moreover, we proposed a GA-based approach to autonomously and systematically set the various parameters of our network architecture. It has been shown that using the proposed GA-based approach increases the accuracy compared to using the manually set parameters. Results have been presented for 9-label, 26-label and 29-label classification. The proposed method has provided promising results, achieving an overall accuracy of 86.6% 77.2%, and 83.03% for 9-label, 26-label and 29-label classification, respectively. This approach can be readily extended to classify more activity types. As future work, we will incorporate a generative adversarial network-based approach to increase the range of parameters that can be chosen autonomously.

TABLE IV: Accuracy rates from different modalities and approaches for 26-label classification

Sensor Modality	Method	Accuracy
IMU only	LSTM	49.2%
Camera only	VGG16	57.7%
	CapsNet	52%
Camera and IMU	VGG16 & LSTM	74.6%
	RecCapsNet & LSTM (Proposed)	**77.2%**

TABLE V: Comparison of different approaches

Method	Sensor Modality	No. of classes	Accuracy
Soran et al. [45]	ego. cam.	28	37.92%
Ours	ego. cam	26	**52%**
Ours	ego. cam & IMU	26	77.2%
Spriggs et al. [32]	ego. cam & IMU	29	57.8%
Ours	ego. cam & IMU	29	**83.03%**

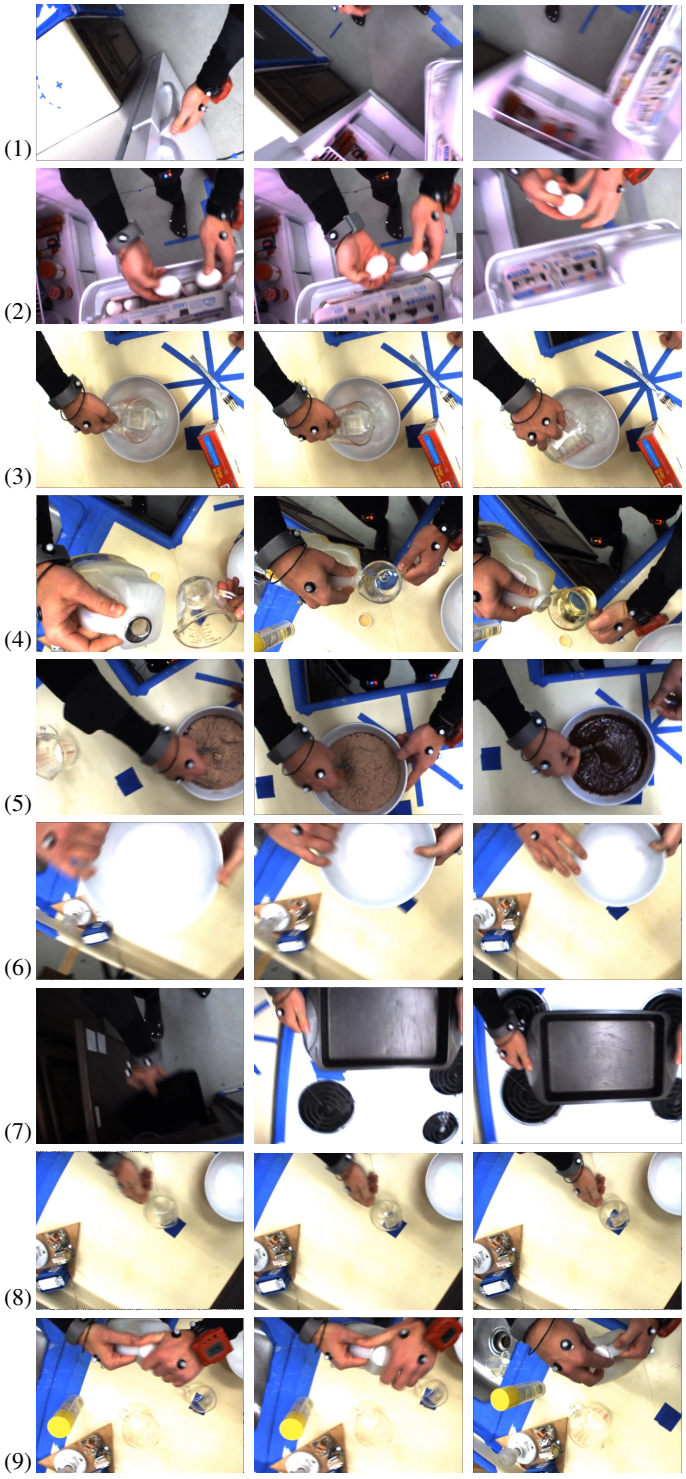


Fig. 9: Example images of the 9 activity classes from the CMU-MMAC dataset. Rows: (1) using fridge, (2) taking eggs, (3) pouring into big bowl, (4) pouring into a measuring cup, (5) stirring in a big bowl, (6) taking bowl, (7) taking baking pan, (8) taking measuring cup, (9) twisting cap (on or off). Columns (a), (b) and (c) show images from the beginning, middle and end of each activity.

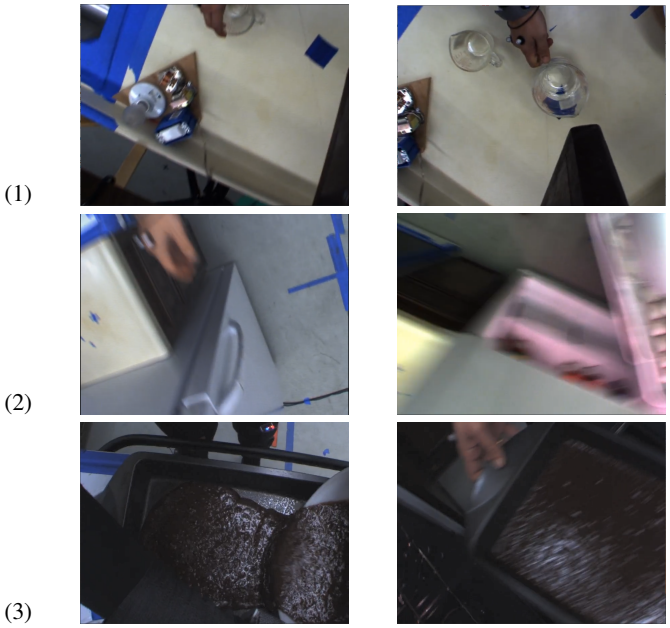


Fig. 10: Examples images of challenging cases causing confusion. Rows: (1) taking a small cup (on the left) vs. big cup (on the right), (2) walking to fridge (on the left) vs. closing fridge (on the right), (3) pouring into pan (on the left) vs. putting the pan into oven (on the right).

REFERENCES

[1] A. Mannini and A. M. Sabatini, "Machine learning methods for classifying human physical activity from on-body accelerometers," Sensors, vol. 10, no. 2, pp. 1154–1175, 2010.

[2] ——, "Accelerometry-based classification of human activities using markov modeling," Intell. Neuroscience, vol. 2011, pp. 4:1–4:10, Jan. 2011. [Online]. Available: http://dx.doi.org/10.1155/2011/647858

[3] F. Attal, S. Mohammed, M. Dedabrishvili, F. Chamroukhi, L. Oukhellou, and Y. Amirat, "Physical human activity recognition using wearable sensors," Sensors, vol. 15, no. 12, pp. 31 314–31 338, 2015.

[4] N. Abhayasinghe and I. Murray, "Human activity recognition using thigh angle derived from single thigh mounted imu data," in Proc. of Int'l Conf. on Indoor Positioning and Indoor Navigation (IPIN), Oct 2014, pp. 111–115.

[5] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," ACM Comput. Surv., vol. 46, no. 3, pp. 33:1–33:33, Jan. 2014.

[6] M. Zhang and A. A. Sawchuk, "Motion primitive-based human activity recognition using a bag-of-features approach," in Proc. of the 2Nd ACM SIGHIT International Health Informatics Symposium, ser. IHI '12. New York, NY, USA: ACM, 2012, pp. 631–640.

[7] A. Bayat, M. Pomplun, and D. A. Tran, "A study on human activity recognition using accelerometer data from smartphones," Procedia Computer Science, vol. 34, pp. 450 – 457, 2014, the 9th Int'l Conf. on Future Networks and Communications (FNC'14)/The 11th Int'l Conf. on Mobile Systems and Pervasive Computing (MobiSPC'14)/Affiliated Workshops.

[8] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," Sensors, vol. 16, no. 115, 2016.

[9] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F. Li, "Large-scale video classification with convolutional neural networks," in Computer Vision and Pattern Recognition, 2014, pp. 1725–1732.

[10] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko, "Long-term recurrent convolutional networks for visual recognition and description," in Comp. Vision and Pattern Recogn., 2015, pp. 677–691.

[11] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), ser. ICCV '15. Washington, DC, USA: IEEE Computer Society, 2015, pp. 4489–4497.

[12] A. Montes, A. Salvador, S. Pascual, and X. Giroinieto, "Temporal

activity detection in untrimmed videos with recurrent neural networks," arXiv:1608.08128, 2017.
[13] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. C. Niebles, "Sst: Single-stream temporal action proposals," in IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), July 2017, pp. 6373–6382.
[14] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," CoRR, vol. abs/1212.0402, 2012. [Online]. Available: http://arxiv.org/abs/1212.0402
[15] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in Computer Vision and Pattern Recognition, 2015, pp. 961–970.
[16] L. Song, Y. Wang, J.-J. Yang, and J. Li, "Health sensing by wearable sensors and mobile phones: A survey," in e-Health Networking, Applications and Services (Healthcom), IEEE Int'l Conf. on, Oct 2014, pp. 453–459.
[17] M. Cornacchia, K. Ozcan, Y. Zheng, and S. Velipasalar, "A survey on activity detection and classification using wearable sensors," IEEE Sensors Journal, vol. 17, no. 2, pp. 386–403, Jan 2017.
[18] C. Li and K. M. Kitani, "Pixel-level hand detection in ego-centric videos," in IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3570–3577.
[19] Y. J. Lee, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization," in Computer Vision and Pattern Recognition, 2012, pp. 1346–1353.
[20] M. S. Ryoo and L. Matthies, "First-person activity recognition: What are they doing to me?" in IEEE Conf. on CVPR, 2013, pp. 2730–2737.
[21] A. Fathi, X. Ren, and J. M. Rehg, "Learning to recognize objects in egocentric activities," in IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 3281–3288.
[22] K. M. Kitani, T. Okabe, Y. Sato, and A. Sugimoto, "Fast unsupervised ego-action learning for first-person sports videos," in Computer Vision and Pattern Recognition, 2011, pp. 3241–3248.
[23] H. Pirsiavash and D. Ramanan, "Detecting activities of daily living in first-person camera views," in 2012 IEEE Conference on Computer Vision and Pattern Recognition, June 2012, pp. 2847–2854.
[24] T. McCandless and K. Grauman, "Object-centric spatio-temporal pyramids for egocentric activity recognition," in BMVC, 2013, pp. 30.1–30.11.
[25] Y. Li, A. Fathi, and J. M. Rehg, "Learning to predict gaze in egocentric video," in IEEE Int'l Conf. on Computer Vision, 2014, pp. 3216–3223.
[26] M. Moghimi, P. Azagra, L. Montesano, and A. C. Murillo, "Experiments on an rgb-d wearable vision system for egocentric activity recognition," in Computer Vision and Pattern Recognition Workshops, 2014, pp. 611–617.
[27] Z. Lu and K. Grauman, "Story-driven summarization for egocentric video," in Computer Vision and Pattern Recognition, 2013, pp. 2714–2721.
[28] T. H. Nguyen, J. C. Nebel, and F. Florezrevuelta, "Recognition of activities of daily living with egocentric vision: A review," Sensors, vol. 16, no. 1, p. 72, 2016.
[29] F. Conti, D. Palossi, R. Andri, M. Magno, and L. Benini, "Accelerated visual context classification on a low-power smartwatch," IEEE Trans. on Human-Machine Systems, vol. 48, no. 1, pp. 19–30, 2017.
[30] K. Zhan, S. Faux, and F. Ramos, "Multi-scale conditional random fields for first-person activity recognition on elders and disabled patients," Pervasive and Mobile Computing, vol. 16, Part B, pp. 251–267, 2015, selected Papers from the 12th Annual {IEEE} Int'l Conf. on Pervasive Computing and Communications (PerCom 2014).
[31] J. Windau and L. Itti, "Situation awareness via sensor-equipped eyeglasses," in 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nov 2013, pp. 5674–5679.
[32] E. H. Spriggs, F. D. L. Torre, and M. Hebert, "Temporal segmentation and activity classification from first-person sensing," in Computer Vision and Pattern Recogn. Workshops. IEEE Computer Society Conf. on, 2009, pp. 17–24.
[33] Y. Lu and S. Velipasalar, "Human activity classification from wearable devices with cameras," in Signals, Systems, and Computers, Asilomar Conf. on, Oct 2017.
[34] F. Torre, J. Hodgins, A. Bargteil, X. Martin, J. Macey, A. Collado, and P. Beltran, "Guide to the carnegie mellon university multimodal activity (cmu-mmac) database," in Tech. report CMU-RI-TR-08-22, Robotics Institute, Carnegie Mellon University, April 2008.
[35] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in Advances in Neural Information Processing Systems 30: Annual Conf. on NIPS, 2017, pp. 3859–3869.
[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.
[37] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in European conference on computer vision. Springer, 2014, pp. 818–833.
[38] M. Lin, Q. Chen, and S. Yan, "Network in network," arXiv preprint arXiv:1312.4400, 2013.
[39] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich et al., "Going deeper with convolutions," in 2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), June 2015, pp. 1–9.
[40] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, 2016, pp. 2818–2826.
[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[42] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in Advances in neural information processing systems, 2015, pp. 91–99.
[43] Y. Lu and S. Velipasalar, "Human activity classification incorporating egocentric video and inertial measurement unit data," in Signal and Information Processing (GlobalSIP), IEEE Global Conf. on, Nov. 2018.
[44] X. Shi, Z. Gao, L. Lausen, H. Wang, D. Yeung, W. Wong, and W. Woo, "Deep learning for precipitation nowcasting: A benchmark and A new model," in Advances in Neural Information Processing Systems 30: Annual Conf. on NIPS, 2017, pp. 5622–5632.
[45] B. Soran, A. Farhadi, and L. Shapiro, "Action recognition in the presence of one egocentric and multiple static cameras," in ACCV 2014. Lecture Notes in Computer Science,, 2014, pp. 17–24.
[46] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," CoRR, vol. abs/1409.1556, 2014.

Yantao Lu (M'16) received the B.S. degree in electrical engineering from Xian Jiaotong University, China in 2013 and the M.S. degree in electrical engineering from Syracuse University, Syracuse, NY, USA in 2015. He is currently working towards the Ph.D. degree in the Department of Electrical Engineering and Computer Science at Syracuse University. His research interests include activity recognition from wearable cameras, and smart and mobile camera systems.

Senem Velipasalar (M'04–SM'14) received the B.S. degree in electrical and electronic engineering from Bogazici University, Istanbul, Turkey, in 1999, the M.S. degree in electrical sciences and computer engineering from Brown University, Providence, RI, USA, in 2001, and the M.A. and Ph.D. degrees in electrical engineering from Princeton University, Princeton, NJ, USA, in 2004 and 2007, respectively. From 2001 to 2005, she was with the Exploratory Computer Vision Group, IBM T. J. Watson Research Center, NY, USA. She is an Associate Professor in the Department of Electrical Engineering and Computer Science at Syracuse University. The focus of her research has been on mobile camera applications, wireless embedded smart cameras, multi-camera tracking and surveillance systems, and automatic event detection from videos.

Dr. Velipasalar received a Faculty Early Career Development Award (CAREER) from the National Science Foundation in 2011. She is the recipient of the 2014 Excellence in Graduate Education Faculty Recognition Award. She is the coauthor of the paper, which received the 2017 IEEE Green Communications and Computing Technical Committee Best Journal Paper Award. She received the Best Student Paper Award at the IEEE International Conference on Multimedia and Expo in 2006. She is the recipient of the EPSCoR First Award, IBM Patent Application Award, and Princeton and Brown University Graduate Fellowships. She is a member of the Editorial Board of the IEEE Transactions on Image Processing and Springer Journal of Signal Processing Systems.